

## IN THE SPECIFICATION

Please amend the paragraph at page 3, lines 17-24, as follows:

Solution 6 therefore seems of interest. Nevertheless, this solution has the drawback of requiring pre-encoding, as specified in the document entitled "*Multiple parallel concatenation of circular recursive systematic codes*" cited above. The duration of pre-encoding is ~~a not~~ not an insignificant constraint. This ~~time~~ duration is the main factor in the latency of the encoder, that is to say the delay between the inputting of a first bit into the encoder and the outputting of a first encoded bit. This is a particular nuisance for certain applications sensitive to transmission times.

Please amend the Abstract as follows:

For encoding a source sequence of symbols ( $\underline{u}$ ) as an encoded sequence~~[[;]]~~, the source sequence ( $\underline{u}$ ) is divided ~~[[; the]]~~ ~~[[ (508) ]]~~ into  $p_1$  first sub-sequences ( $\underline{U}_i$ ),  $p_1$  being a positive integer, and each of the first sub-sequences ( $\underline{U}_i$ ) is encoded ~~by means of~~ in a first circular convolutional encoding method. ~~[[; the]]~~ The source sequence ( $\underline{u}$ ) is interleaved ~~[[ (506) ]]~~ into an interleaved sequence ( $\underline{u}^*$ )~~[[;]]~~, and the interleaved sequence ( $\underline{u}^*$ ) is divided ~~[[ (507) ]]~~ into  $p_2$  second sub-sequences ( $\underline{U}'_i$ ),  $p_2$  being a positive integer. ; ~~and each~~ Each of the second sub-sequences ( $\underline{U}'_i$ ) is encoded ~~by means of~~ in a second circular convolutional encoding method. At least one of the integers  $p_1$  and  $p_2$  is strictly

greater than 1 and at least one of the first sub-sequences ( $\underline{U}_i$ ) is not interleaved into any of the second sub-sequences ( $\underline{U}'_j$ ). ~~Figure 5.~~

(It is noted that the above underlining of the following symbols is original, and is meant to be permanent:  $\underline{u}$ ,  $\underline{U}_i$ ,  $\underline{u}^*$ ,  $\underline{U}'_i$ ,  $\underline{U}'_j$ )